Arbitrary Ranking of Defeasible Subsumption

Giovanni Casini[1], Michael Harrison[2], Thomas Meyer[2], and Reid Swan[2]

[1] CSC, Université du Luxembourg, Luxembourg
giovanni.casini@uni.lu
[2] CAIR, University of Cape Town, South Africa
{HRRMIC014,SWNREI001}@myuct.ac.za, tmeyer@cs.uct.ac.za

Abstract. In this paper we propose an algorithm that generalises existing procedures for the implementation of defeasible reasoning in the framework of Description Logics (DLs). One of the well-known approaches to defeasible reasoning, the so-called KLM approach, is based on constructing specific rankings of defeasible information, and using these rankings to determine priorities in case of conflicting information. Here we propose a procedure that allows us to input any possible ranking of the defeasible concept inclusions contained in the knowledge base. We analyse and investigate the forms of defeasible reasoning obtained when conclusions drawn are obtained using these rankings.

1 Introduction

Members of the Description Logic (DL) community have devoted considerable time and effort to the introduction of defeasible forms of reasoning into DLs, especially in the last decade. Among the different proposals [4,5,43,1,2,3,24,30,31,32], [36,39,44], particular attention has been paid to the lifting of the so-called KLM approach (named after its originators, Kraus, Lehmann and Magidor [33,34]) to the level of DLs [10,11,12,18,20,23,25,26,27,40,41,45,37,38]. In some cases, propositional versions of well-known defeasible semantics, such as Rational Closure [18,27,16,14] and Lexicographic Closure [19], have been adapted for the DL case. In others, new forms of entailment have been proposed [22,20,21]. Most of the methods based on the KLM approach operate by imposing a ranking on the provided defeasible statements. This ranking is used to determine which pieces of information should override the others in case of any conflict. In this paper we define and investigate the properties of a decision procedure that generalises this approach by allowing for an arbitrary ranking to be used as input. It is worth noting that the Rational Closure and Lexicographic Closure methods are special cases of our proposal.

The main contributions of this paper are showing that, by repurposing an existing defeasible entailment algorithm, it is possible to reason with an arbitrary ranking of defeasible subsumption statements while satisfying the KLM postulates; we provide equivalence and redundancy results of arbitrary rankings, and an optimization procedure that puts these to use to reduce redundant computation; and show that for any arbitrary ranking, there is a defeasible knowledge base the Rational Closure of which is equivalent to that arbitrary ranking. The results are significant, as they allow for the alteration or repair of automatically generated rankings (for example, those generated by lexicographic or rational closure) in the case that undesired statements are entailed, without sacrificing any of the KLM postulates when doing so. This should provide ontology engineers with greater control over their knowledge bases.

2 Background

We assume that the reader is familiar with the basic notions of DLs. We base our work on the DL $\mathcal{ALC}$. Given a finite set of atomic *concept names* $\mathsf{C} := \{A, B, \ldots\}$ and a finite set of *role names* $\mathsf{R} := \{r, s, \ldots\}$, the set of complex concepts is defined as follows: $\mathcal{C} := \top \mid \bot \mid \mathsf{C} \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall r.C \mid \exists r.C$. For $C, D \in \mathsf{C}$, a *General Concept Inclusion* (*GCI*) has the form $C \sqsubseteq D$, with a *TBox* $\mathcal{T}$ being a finite set of GCIs. The expression $C \equiv D$ is an abbreviation of $C \sqsubseteq D$ and $D \sqsubseteq C$. $\mathcal{T} \models C \sqsubseteq D$ denotes that $C \sqsubseteq D$ is classically entailed by $\mathcal{T}$. Following previous proposals [10,18] a *Defeasible Concept Inclusion* (*DCI*) has the form $C \sqsubseteq\!\!\!\sim D$ and is read as "Usually, an element of C is also an element of D". A DTBox (Defeasible TBox) is a finite set of DCIs.

Example 1. Suppose we know that Mammals generally are terrestrial, but whales are mammals and marine creatures. Using classical GCIs we would have a TBox as $\mathcal{T} = \{\mathtt{Mammal} \sqsubseteq \mathtt{LandDweller}, \mathtt{Whale} \sqsubseteq \mathtt{Mammal}, \mathtt{Whale} \sqsubseteq \mathtt{WaterDwellers}, \mathtt{WaterDwellers} \sqsubseteq \neg\mathtt{LandDwellers}\}$, that would classically entail that whales do not exist ($\mathcal{T} \models \mathtt{Whale} \sqsubseteq \bot$). Using DCIs we can convey the information above more accurately, eliminating $\mathtt{Mammal} \sqsubseteq \mathtt{LandDweller}$ from the TBox and adding a DTBox $\mathcal{D} = \{\mathtt{Mammal} \sqsubseteq\!\!\!\sim \mathtt{LandDweller}\}$, indicating that Mammals *generally* are terrestrial, but leaving open the possibility for the existence of exceptions (like whales).

Consider knowledge bases of the form $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. We focus here on *defeasible entailment* (denoted by $\models\!\!\!\approx$). That is, we are interested in what it means for GCIs and DCIs to be entailed by a knowledge base $\mathcal{K}$. It is well-accepted that defeasible entailment (unlike classical entailment) is not unique. In studying different forms of defeasible entailment, the position we adopt here is to consider the *rationality properties* presented in Fig. 2 that any form of defeasible entailment ought to satisfy. These are referred to as the KLM properties, and are translations of the propositional properties initially proposed by Kraus et al. [33,34] into a DL setting.

A number of forms of defeasible entailment have been defined that satisfy all the KLM properties [18,27,19,20,21,8], with Rational Closure being perhaps the best known of these. Originally proposed by Lehmann and Magidor for the propositional case [34], it has been redefined for the DL case [18,27,8]. Rational Closure implements a desirable principle of reasoning about expectations, the *presumption of typicality*. One school of thought has it that all interesting forms of defeasible entailment should extend Rational Closure [17]. The principle of

$$\mathcal{K} \models\!\approx C \precsim C \ \text{(Ref)} \qquad \frac{C \equiv D,\ \mathcal{K} \models\!\approx C \precsim E}{\mathcal{K} \models\!\approx D \precsim E} \ \text{(LLE)}$$

$$\frac{\mathcal{K} \models\!\approx C \precsim D,\ \mathcal{K} \models\!\approx C \precsim E}{\mathcal{K} \models\!\approx C \precsim D \sqcap E} \ \text{(And)} \qquad \frac{\mathcal{K} \models\!\approx C \precsim E,\ \mathcal{K} \models\!\approx D \precsim E}{\mathcal{K} \models\!\approx C \sqcup D \precsim E} \ \text{(Or)}$$

$$\frac{\mathcal{K} \models\!\approx C \precsim D,\ D \sqsubseteq E}{\mathcal{K} \models\!\approx C \precsim E} \ \text{(RW)} \qquad \frac{\mathcal{K} \models\!\approx C \precsim D,\ \mathcal{K} \models\!\approx C \precsim E}{\mathcal{K} \models\!\approx C \sqcap E \precsim D} \ \text{(CM)}$$

$$\frac{\mathcal{K} \models\!\approx C \precsim D,\ \mathcal{K} \not\models\!\approx C \precsim \neg E}{\mathcal{K} \models\!\approx C \sqcap E \precsim D} \ \text{(RM)}$$

Fig. 1. The KLM postulates

presumption of typicality states that we reason assuming that we are in the most typical situation consistent with the information at our disposal. For example, if we *only* know that manatees are mammals (`Manatee` $\sqsubseteq$ `Mammal`) and that usually mammals are land dwellers (`Mammal` $\precsim$ `LandDweller`), we assume that manatees are normal mammals, and we would like to conclude that *presumably* they are land dwellers (`Manatee` $\precsim$ `LandDweller`). The *presumption of typicality* is clearly a defeasible form of reasoning. If we are later informed that manatees live in the water (`Manatee` $\precsim$ `WaterDweller`), then we want to drop the previously tentative conclusion that manatees are land dwellers (`Manatee` $\precsim$ `LandDweller`).

Rational Closure can be defined semantically in the DL framework [27,23,8]. For our purposes in this paper, though, it is sufficient to focus on the description of Rational Closure in terms of a decision procedure. This procedure relies on a series of classical $\mathcal{ALC}$ entailment checks, and preserves the computational complexity of classical $\mathcal{ALC}$ entailment checking [18,8]. Given $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, the basic step in the procedure is to determine whether an element of $\mathcal{D}$ is *exceptional* w.r.t. $\mathcal{K}$. Informally, a DCI $C \precsim D$ is exceptional w.r.t. a KB $\mathcal{K}$ if C represents a class satisfying atypical properties. Consider Example 1, and add to $\mathcal{D}$ the DCI `Whale` $\precsim \neg$`Aggressive`. The latter is an exceptional DCI w.r.t. $\mathcal{K}$, since whales are marine creatures and are exceptional w.r.t. the class of mammals, that are usually land dwellers.

Given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, Algorithm 1 returns the elements of $\mathcal{D}$ that are exceptional. To do that, the *materialization* of a set of DCIs is used.

Definition 1 (Materialization). *The* materialization *of a set of DCIs $\mathcal{D}$, denoted by $\overline{\mathcal{D}}$, is defined as $\overline{\mathcal{D}} := \{\neg C \sqcup D \mid C \precsim D \in \mathcal{D}\}$.*

That is, the materialisation of a DCI $C \precsim D$ is the concept $\neg C \sqcup D$, representing the corresponding material implication.

Using Algorithm 1, we obtain an algorithm (Algorithm 2) which, given $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, *ranks* all the DCIs in $\mathcal{D}$ according to their level of exceptionality. In general, in case of conflictual information (e.g., students do not pay taxes, but working students do), the subconcepts that present atypical properties, and the DCIs that have as antecedents such subconcepts, have a higher rank: students,

Algorithm 1 Exceptional

procedure EXCEPTIONAL
  Input: $\mathcal{K} = (\mathcal{T}, \mathcal{D})$
  Output: $\mathcal{D}' \subseteq \mathcal{D}$ such that $\mathcal{D}'$ is exceptional w.r.t. $\mathcal{D}$
  for $A \sqsubset\!\!\!\sim B \in \mathcal{D}$ **do**
    if $\mathcal{T} \models \bigsqcap \overline{\mathcal{D}} \sqsubseteq \neg A$ **then**
      $\mathcal{D}' := \mathcal{D}' \cup \{A \sqsubset\!\!\!\sim B\}$
  return $\mathcal{D}'$

that typically do not pay taxes, will have rank 0; working students, since they pay taxes, will have rank 1; let there be a rule stating that working students with a family do not have to pay taxes, and so they will have rank 2, and so on.

Algorithm 2 ComputeRanking

procedure COMPUTERANKING
  Input: $\mathcal{K} = (\mathcal{T}, \mathcal{D})$
  Output: $\mathcal{K}' = (\mathcal{T}', \mathcal{D}')$ and a ranking $\mathcal{E}$ of $\mathcal{D}$.
  $\mathcal{T}' := \mathcal{T}$, $\mathcal{D}' := \mathcal{D}$, $\mathcal{E} := \emptyset$, $\mathcal{D}'_\infty := \mathcal{D}$
  while $\mathcal{D}'_\infty \neq \emptyset$ **do**
    i := 0
    $\mathcal{E}_0 := \mathcal{D}'$
    $\mathcal{E}_1 :=$ Exceptional$(\mathcal{T}', \mathcal{E}_0)$
    while $\mathcal{E}_{i+1} \neq \mathcal{E}_i$ **do**
      $i := i + 1$
      $\mathcal{E}_{i+1} :=$ Exceptional$(\mathcal{T}', \mathcal{E}_i)$
    $\mathcal{E} := (\mathcal{E}_0, \ldots, \mathcal{E}_{i-1})$
    $\mathcal{D}'_\infty := \mathcal{E}_i$
    $\mathcal{T}' := \mathcal{T}' \cup \{C \sqsubseteq D \mid C \sqsubset\!\!\!\sim D \in \mathcal{D}'_\infty\}$
    $\mathcal{D}' := \mathcal{D}' \backslash \mathcal{D}'_\infty$
  $\mathcal{K}' := (\mathcal{T}', \mathcal{D}')$
  return $(\mathcal{K}', \mathcal{E})$

In addition to the ranking $\mathcal{E}$ that Algorithm 2 produces, it also outputs a KB $\mathcal{K}'$ that, despite not necessarily being the same as the original $\mathcal{K}$, is equivalent to $\mathcal{K}$ [8, Lemma 13]. We refer the reader to the work of Britz et al. [8] for an explanation of the reasons behind the introduction of $\mathcal{K}'$, as well as for a more detailed discussion of the `ComputeRanking` algorithm.

The next algorithm we consider, Algorithm 3, takes as input a KB $\mathcal{K}$ and a ranking $\mathcal{E}$, as well as a query $C \sqsubset\!\!\!\sim D$, and decides whether $C \sqsubset\!\!\!\sim D$ is defeasibly entailed by $\mathcal{K}$ (given $\mathcal{E}$). Algorithm 3 guarantees that, in case of potential conflicts, the information that is higher in the ranking overrides the other pieces of information. For example, if students do not pay takes (`Student` $\sqsubset\!\!\!\sim$ $\neg$`PayTax`) has rank 0, while the rule that working students should pay taxes (`Student` $\sqcap$ `Worker` $\sqsubset\!\!\!\sim$ `PayTax`) has rank 1, we will conclude that, for example, part-time students that have a job as public servants presumably should pay taxes (`PartTimeStudent` $\sqcap$ `PubServ` $\sqsubset\!\!\!\sim$ `PayTax`), since, in case of conflict, we rely on the DCI with the higher rank.

Finally, combining Algorithm 3 with Algorithm 2 enables us to define an algorithm for computing Rational Closure [8, Theorem 4]: given a KB $\mathcal{K}$ and a query $C \mathrel{\sqsubset\!\!\!\sim} D$, Algorithm 4 takes the ranking created by Algorithm 2 from $\mathcal{K}$ and, calling Algorithm 3, decides whether $C \mathrel{\sqsubset\!\!\!\sim} D$ is or not in the Rational Closure of $\mathcal{K}$. DIP[***] (Defeasible Inference Platform) is a Protégé plugin implementing Rational Closure for DLs [14,16].

Algorithm 3 IsEntailed

procedure IsEntailed
  Input: $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$, query $C \mathrel{\sqsubset\!\!\!\sim} D$
  Output: `true` iff $C \mathrel{\sqsubset\!\!\!\sim} D$ is defeasibly entailed by $\mathcal{K}$
  i := 0
  while $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap C \sqsubseteq \bot$ and $i \leq n$ **do**
    $i := i + 1$
  if $i \leq n$ **then**
    return $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap C \sqsubseteq D$
  else
    return $\mathcal{T} \models C \sqsubseteq D$

Algorithm 4 RationalClosure

procedure RationalClosure
  Input: $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, query $C \mathrel{\sqsubset\!\!\!\sim} D$
  Output: `true` iff $C \mathrel{\sqsubset\!\!\!\sim} D$ is in the Rational Closure of $\mathcal{K}$.
  $(\mathcal{K}', \mathcal{E}) := ComputeRanking(\mathcal{K})$
  return IsEntailed($\mathcal{K}', \mathcal{E}, C \mathrel{\sqsubset\!\!\!\sim} D$)

3 Rankings and Ranking Equivalence

As previously noted, Algorithm 2 returns a fixed ranking $\mathcal{E} = (\mathcal{E}_0, \ldots, \mathcal{E}_n)$ when presented with a KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ as input. That is, it orders the elements of $\mathcal{D}$ w.r.t. exceptionality. In what follows we will refer to a more general notion of ranking of a KB.

Definition 2 (Ranking). *A ranking is a tuple $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_n)$ defined relative to a KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ satisfying the following properties:*

- *for every integer $i \in [0, n]$, $\mathcal{E}_i \subseteq \mathcal{D}$;*
- *for every integer $i \in [0, n)$, $\mathcal{E}_{i+1} \subseteq \mathcal{E}_i$*
- *$\mathcal{E}_0 = \mathcal{D}$*

A *rank* refers to any of the sets which compose a given ranking $\mathcal{E}$. For all integers $i \in [0, n]$, $\mathcal{E}_i$ is a rank of $\mathcal{E}$. The *size* of $\mathcal{E}$, denoted $|\mathcal{E}|$, is n where $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_n)$.

A property of the materialization of ranks of $\mathcal{E}$ is shown in the following lemma.

[***] https://tinyurl.com/y3cjdznp

Lemma 1. *Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ be a ranking relative to the KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. For every integer $i \in [0, n)$, $\models \bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bigsqcap \overline{\mathcal{E}_{i+1}}$.*

It follows that if $0 \leq j < i \leq n$, then $\models \bigsqcap \overline{\mathcal{E}_j} \sqsubseteq \bigsqcap \overline{\mathcal{E}_i}$.

What happens if, together with a KB $\mathcal{K}$, with give as input to Algorithm 3 an arbitrary ranking $\mathcal{E}$ of $\mathcal{K}$ instead of the ranking produced by `ComputeRanking`? It turns out that `IsEntailed` still defines an entailment relation that satisfies all the KLM properties. To make this more precise, we define defeasible entailment w.r.t a ranking $\mathcal{E}$ as follows.

Definition 3. *Given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ and an arbitrary ranking $\mathcal{E}$ of $\mathcal{K}$, we say that $C \sqsubsim D$ is defeasibly entailed by $\mathcal{K}$ w.r.t. $\mathcal{E}$, denoted by $\mathcal{K} \mid\!\approx_{\mathcal{E}} C \sqsubsim D$, if* `IsEntailed`$(\mathcal{K}, \mathcal{E}, C \sqsubsim D) =$ `true`

Observe that defeasible entailment for $\mathcal{K}$ w.r.t the ranking produced by `ComputeRanking` is exactly Rational Closure.

Theorem 1. *For a KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ and a ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ of $\mathcal{K}$, defeasible entailment for $\mathcal{K}$ w.r.t $\mathcal{E}$ ($\mid\!\approx_{\mathcal{E}}$) satisfies all the KLM properties.*

To motivate reasoning with an arbitrary ranking, note that algorithms like Algorithm 2 determine a ranking of statements based on their logical specificity, wherein more specific statements are ranked higher; in some cases, we may wish to base our rankings on another principle altogether. Consider the example of country C. C is federal, with every region having a regional code, which is superceded by the federal code of C. Alcohol is considered a narcotic substance, but in region R of C, alcohol consumption is permitted. New laws are introduced to the federal code of C: one states that the consumption of narcotics is a crime unless otherwise indicated in the federal code, while the other states that consumption of narcotics is permitted if medically prescribed. Citizen B is caught drinking grappa in region R, without a medical prescription. To model that Bob is guilty of having broken the law of C, all that is required in the arbitrary ranking framework is to give the laws of the federal code a lower ranking than those of the regional code.

We now investigate the circumstances under which two rankings entail the same set of statements, that is, when two rankings are *equivalent*. Results about the equivalence of rankings are of interest, because as per Lutz [35], entailment checking in $\mathcal{ALC}$ is ExpTime-complete, and since the algorithms for defeasible entailment used in this paper (in particular, `IsEntailed`) reduce to linearly many $\mathcal{ALC}$ entailment checks, reducing the number of entailment checks required can have a significant impact on the running time of these algorithms. Thus finding an equivalent knowledge base of a smaller size may allow faster run times without sacrificing reasoning power.

Definition 4 (Equivalence). *Two rankings $\mathcal{E}$, $\mathcal{F}$, relative to the KBs $\mathcal{K}$ and $\mathcal{K}'$ respectively, are* equivalent *($\mathcal{E} \doteq \mathcal{F}$) if, for every DCI $C \sqsubsim D$, $\mathcal{K} \mid\!\approx_{\mathcal{E}} C \sqsubsim D$ if and only if $\mathcal{K}' \mid\!\approx_{\mathcal{F}} C \sqsubsim D$.*

Clearly $\doteq$ is an equivalence relation.

Lemmas 2 and 3 below allow us to 'shrink' a ranking in order to produce a ranking which is smaller but equivalent. For notational convenience, we define *subrankings*, borrowing syntax from the Python programming language.

Definition 5 (Subranking). *Let $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_n)$ be a ranking relative to $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. For $0 \leq i \leq j \leq n$, the tuple $(\mathcal{E}_i, \mathcal{E}_{i+1}, ..., \mathcal{E}_j)$ is an i-to-j* subranking *of $\mathcal{E}$, denoted $\mathcal{E}[i : j]$.*

Lemma 2. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ be a KB with the statements in $\mathcal{D}$ ranked into $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$. If there is some $i \in [0..n]$ for which $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqsubseteq \bot$ then $\mathcal{E} \doteq \mathcal{E}[i+1 : n]$, where $\mathcal{E}[i+1 : n]$ is the subranking of $\mathcal{E}$ relative to KB $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ where $\mathcal{D}' = \mathcal{E}_{i+1}$.*

Lemma 3. *Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ be a ranking relative to KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. Let $i \in [0, n]$ be the smallest integer such that $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \equiv \top$. Then $\mathcal{E}' \doteq \mathcal{E}$, where $\mathcal{E}' = \mathcal{E}[0 : i-1]$.*

These lemmas allow for ranks to be dropped from the top or bottom of a ranking without changing the set of entailed statements, but what about dropping statements from the middle of the ranking? This is discussed further in Section 4, but the following definition allows us to provide some initial results:

Definition 6 (Dense ranking). *Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ be a ranking. $\mathcal{E}$ is dense if for all $0 \leq i < n$, $\mathcal{T} \not\models \bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{E}_{i+1}}$.*

Dense rankings are significant because every rank of a dense ranking is involved in deciding entailment. By contrast, a ranking which is not dense (a *sparse* ranking) has the property that for some $\mathcal{E}_i$, $\mathcal{E}_{i+1}$, $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{E}_{i+1}}$. For any concept A, $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \equiv \bigsqcap \overline{\mathcal{E}_{i+1}} \sqcap A$; so A is non-empty at i if and only if it is non-empty at $i+1$. If $\bigsqcap \overline{\mathcal{E}_i} \sqcap A$ is non-empty then $\bigsqcap \overline{\mathcal{E}_{i+1}} \sqcap A$ is non-empty, but algorithm 3 will stop at i, so $\mathcal{E}_{i+1}$ will not be used; and if $\bigsqcap \overline{\mathcal{E}_i} \sqcap A$ is empty, then $\bigsqcap \overline{\mathcal{E}_{i+1}} \sqcap A$ is also empty, so it will be passed over for the next highest rank. $\mathcal{E}_{i+1}$ therefore never contributes to entailment checking. This can be thought of as a 'gap' in the ranking that does not alter entailment in a significant way, increasing the running time of entailment checking without contributing knowledge.

To this end, algorithm 5, `CollapseRanking`, takes a KB $\mathcal{K}$ and ranking $\mathcal{E}$ and returns a ranking $\mathcal{E}'$ of the same statements such that $\mathcal{E}'$ is dense and $\mathcal{E}' \doteq \mathcal{E}$. Note that the algorithm performs linearly many $\mathcal{ALC}$ entailment checks in the size of the ranking. Schmidt-Schauß and Smolka prove in [42] that $\mathcal{ALC}$ entailment checking is PSPACE-complete, and provide an algorithm for entailment checking in linear space and exponential time. `CollapseRanking` therefore should have no worse than exponential running time in the size of the ranking.

Algorithm 5 CollapseRanking

procedure $\textsc{CollapseRanking}(\mathcal{K}, \mathcal{E})$
  Input: $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, finite ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$
  Output: $\mathcal{E}'$ where $\mathcal{E}'$ is dense and equivalent to $\mathcal{E}$.
  $\mathcal{E}'_0 = \mathcal{E}_0$
  $i := 1$
  $j := 1$
  $\texttt{prev} := \bigsqcap \overline{\mathcal{E}_0}$
  while $i \leq n$ **do**
    if $\mathcal{T} \not\models \bigsqcap \overline{\mathcal{E}_i} \equiv \texttt{prev}$ **then**
      $\mathcal{E}'_j := \mathcal{E}_i$
      $\texttt{prev} := \bigsqcap \overline{\mathcal{E}_i}$
      $j := j + 1$
    $i := i + 1$
  return $\mathcal{E}'$

Theorem 2. *The ranking* $\mathcal{E}' = \texttt{CollapseRanking}(\mathcal{K}, \mathcal{E})$ *is dense and equivalent to* $\mathcal{E}$.

Corollary 1. $\mathcal{E} \doteq \mathcal{F}$ *if and only if* $\texttt{CollapseRanking}(\mathcal{E}) \doteq \texttt{CollapseRanking}(\mathcal{F})$.

Corollary 2. *Every finite ranking* $\mathcal{E}$ *has an equivalent dense ranking.*

Another motivation for working with dense rankings is the result of Theorem 3 below, which states that if two dense rankings are equivalent, then the conjunction of the materialization of each of their ranks must be equivalent.

One requirement of the proof of Theorem 3 is that the top rank of the rankings under consideration not have the conjunction of its materialization equivalent to $\top$. But by Lemma 3, it is always possible to transform a ranking into an equivalent ranking for which this is true.

Lemma 4 is split from Theorem 3 because it allows us to prove a slighter stronger result that does not require dense rankings.

Lemma 4. *Let* $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_m)$ *and* $\mathcal{F} = (\mathcal{F}_0, ..., \mathcal{F}_n)$ *be rankings relative to the same KB* $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ *with* $m = n$ *such that for all integers* i *with* $0 \leq i \leq n$, $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{F}_i}$*; then* $\mathcal{E} \doteq \mathcal{F}$.

Theorem 3. *Let* $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_m)$ *and* $\mathcal{F} = (\mathcal{F}_0, ..., \mathcal{F}_n)$ *be dense rankings relative to the KBs* $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ *and* $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ *respectively, such that* $\mathcal{T} \not\models \bigsqcap \overline{\mathcal{E}_0} \equiv \bot$, $\mathcal{T} \not\models \bigsqcap \overline{\mathcal{F}_0} \equiv \bot$, $\mathcal{T} \not\models \bigsqcap \overline{\mathcal{E}_m} \equiv \top$, *and* $\mathcal{T} \not\models \bigsqcap \overline{\mathcal{F}_n} \equiv \top$*; then* $\mathcal{E} \doteq \mathcal{F}$ *if and only if* $m = n$ *and for all integers* i *such that* $0 \leq i \leq n$, $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{F}_i}$.

The following observation summarizes the section on equivalence.

Let $\mathcal{E}$ and $\mathcal{F}$ be two rankings of statements relative to KBs $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ and $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ respectively. Let $\mathcal{E}' = \texttt{CollapseRanking}(\mathcal{K}, \mathcal{E})$, $\mathcal{F}' = \texttt{CollapseRanking}(\mathcal{K}', \mathcal{F})$. Let $n = |\mathcal{E}'|$ and $m = |\mathcal{F}'|$. Assign:

$$e_l = \begin{cases} 0, & \mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_0} \not\equiv \bot \\ 1, & \mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_0} \equiv \bot \end{cases} \quad e_h = \begin{cases} n, & \mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_n} \not\equiv \top \\ n-1, & \mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_n} \equiv \top \end{cases}$$

$$f_l = \begin{cases} 0, & \mathcal{T} \models \bigsqcap \overline{\mathcal{F}'_0} \not\equiv \bot \\ 1, & \mathcal{T} \models \bigsqcap \overline{\mathcal{F}'_0} \equiv \bot \end{cases} \quad f_h = \begin{cases} n, & \mathcal{T} \models \bigsqcap \overline{\mathcal{F}'_m} \not\equiv \top \\ n-1, & \mathcal{T} \models \bigsqcap \overline{\mathcal{F}'_m} \equiv \top \end{cases}$$

Let $\mathcal{E}'' = \mathcal{E}'[e_l : e_h]$ and $\mathcal{F}'' = \mathcal{F}'[f_l : f_h]$. Finally, let $n' = |\mathcal{E}''|$ and $m' = |\mathcal{F}''|$.

Corollary 3. *$\mathcal{E} \doteq \mathcal{F}$ if and only if $n' = m'$ and for all integers i such that $0 \leq i \leq n'$, $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}''_i} \equiv \bigsqcap \overline{\mathcal{F}''_i}$.*

Algorithm 6 Equivalent

procedure EQUIVALENT($\mathcal{K}, \mathcal{E}, \mathcal{K}', \mathcal{F}$)
  Input: $\mathcal{K} = (\mathcal{T}, \mathcal{D})$ with finite ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$, $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ with finite ranking $\mathcal{F} = (\mathcal{F}_0, ..., \mathcal{F}_n)$
  Output: `true` if and only if $\mathcal{E} \doteq \mathcal{F}$
  $\mathcal{E}' :=$ `CollapseRanking`$(\mathcal{K}, \mathcal{E})$
  $\mathcal{F}' :=$ `CollapseRanking`$(\mathcal{K}, \mathcal{F})$
  $n := |\mathcal{E}'|$, $m := |\mathcal{F}'|$, $e_l := 0$, $e_h := n$, $f_l = 0$, $f_h = m$
  if $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_0} \equiv \bot$ **then**
    $e_l := e_l + 1$
  if $\mathcal{T} \models \bigsqcap \overline{\mathcal{F}'_0} \equiv \bot$ **then**
    $f_l := f_l + 1$
  if $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_n} \equiv \top$ **then**
    $e_h := e_h - 1$
  if $\mathcal{T} \models \bigsqcap \overline{\mathcal{F}'_m} \equiv \top$ **then**
    $f_h := f_h - 1$
  $\mathcal{E}'' := \mathcal{E}'[e_l : e_h]$, $\mathcal{F}'' := \mathcal{F}'[f_l : f_h]$
  $n' := |\mathcal{E}''|$, $m' = |\mathcal{F}''|$
  if $n' \neq m'$ **then**
    return `false`
  for i from 0 to n' **do**
    if $\mathcal{T} \not\models \bigsqcap \overline{\mathcal{E}''} \equiv \bigsqcap \overline{\mathcal{F}''}$ **then**
      return `false`
  return `true`

Algorithm 6, `Equivalent`, uses Corollary 3 to determine if two rankings are equivalent. The algorithm consists of linearly many entailment checks, and since $\mathcal{ALC}$ entailment checking is EXPTIME-complete [35], the algorithm is in the EXPTIME complexity class.

4 Statement redundancy

Broadly, redundant statements are statements which do not effect the set of entailed statements. In other words, on adding or removing redundant statements, the new and old rankings should be equivalent. In this section,we define and discuss two forms of redundancy, *local* and *total* redundancy.

Definition 7 (Locally redundant). *Let $\mathcal{E}$ be a ranking of a DTBox $\mathcal{D}$ in $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, and let $C \precsim D \in \mathcal{D}$. $C \precsim D$ is locally redundant at i if $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{E}'_i}$ where $\mathcal{E}'_i = (\mathcal{E}_i \setminus \{C \precsim D\})$.*

It follows trivially from the definition that if $C \precsim D$ is locally redundant at i and $\mathcal{E}'_i = (\mathcal{E}_i \setminus \{C \precsim D\})$, then $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \equiv \bigsqcap \overline{\mathcal{E}'_i} \sqcap A$ holds for all concepts A; so for any concept B, $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap A \sqsubseteq B$ iff $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_i} \sqcap A \sqsubseteq B$.

To see the relevance of local redundancy at i, consider $\mathcal{E} = (\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_n)$, a ranking relative to KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, and a defeasible subsumption statement $C \precsim D \in \mathcal{E}_i$ that is locally redundant at i, $i > 0$. Let $R = \{A \precsim B | A \precsim B \in \mathcal{E}_{i-1}$ and $A \precsim B \notin \mathcal{E}_i\}$. Note that $C \precsim D \in \mathcal{E}_i$ and $\mathcal{E}_i \subseteq \mathcal{E}_{i-1}$, so $C \precsim D \notin R$.

$\mathcal{E}_i \cup R = \mathcal{E}_{i-1}$, so $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap \bigsqcap \overline{R} \equiv \bigsqcap \overline{\mathcal{E}_{i-1}}$. By assumption, $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \equiv \bigsqcap \overline{\mathcal{E}'_i}$, so $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}'_i} \sqcap \bigsqcap \overline{R} \equiv \bigsqcap \overline{\mathcal{E}_i} \sqcap \bigsqcap \overline{R}$ and in turn $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_i} \sqcap \bigsqcap \overline{R} \equiv \bigsqcap \overline{\mathcal{E}_{i-1}}$. But $\bigsqcap \overline{\mathcal{E}'_i} \sqcap \bigsqcap \overline{R}$ is the conjunction of the materialization of every $A \precsim B \in \mathcal{E}_{i-1}$ except $C \precsim D$, so it is equivalent to $\bigsqcap \overline{\mathcal{E}_{i-1} \setminus \{C \precsim D\}}$. Then $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_{i-1}} \equiv \bigsqcap \overline{\mathcal{E}_{i-1} \setminus \{C \precsim D\}}$, so $C \precsim D$ is locally redundant at $i - 1$. A simple inductive argument yields:

Theorem 4. *If $C \precsim D$ is locally redundant at i then $C \precsim D$ is locally redundant at j for all $0 \leq j \leq i$.*

We also have the apparently stronger *total* redundancy:

Definition 8 (Totally redundant). *For a ranking $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ and conditional KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, a statement $C \precsim D$ is totally redundant if $\mathcal{E} \doteq \mathcal{E}'$, where $\mathcal{E}' = (\mathcal{E}_0 \setminus \{C \precsim D\}, ..., \mathcal{E}_n \setminus \{C \precsim D\})$ relative to $\mathcal{K} = (\mathcal{T}, \mathcal{D} \setminus \{C \precsim D\})$*

In fact, total and local redundancy are strongly related. Theorem 5 shows that for dense rankings, the two are interdefinable.

Theorem 5. *Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ be a dense ranking relative to KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. Let $i \in [0, n]$ be the largest integer such that $C \precsim D \in \mathcal{E}_i$. Then $C \precsim D$ is totally redundant if and only if it is locally redundant at i.*

5 Relating Rational Closure to Rankings

Clearly rational closure is in the class of entailment relations definable using rankings and Algorithm 3, since it corresponds to using the output of `ComputeRanking` as input ranking for Algorithm 3. But is the class of entailment relations definable using rational closure a proper subset of the class of relations definable from rankings and Algorithm 3? Actually, this is not the case: for an arbitrary ranking $\mathcal{E}$ relative to KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$, it is always possible to create a KB $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ such that $\mathcal{E} \doteq \mathcal{E}'$ where $\mathcal{E}' =$ `ComputeRanking`$(\mathcal{K}')$. Since `RationalClosure` uses the ranking output by `ComputeRanking`, we have for all statements $C \precsim D$, `RationalClosure`$(\mathcal{K}', C \precsim D) =$ `IsEntailed`$(\mathcal{K}, \mathcal{E}, C \precsim D)$.

Theorem 6. *Let $\mathcal{E} = (\mathcal{E}_0, ..., \mathcal{E}_n)$ be a ranking relative to conditional KB $\mathcal{K} = (\mathcal{T}, \mathcal{D})$. Let $\mathcal{D}' = \{\top \precsim \bigsqcap \overline{\mathcal{E}_0}\} \cup \{\neg \bigsqcap \overline{\mathcal{E}_i} \precsim \bigsqcap \overline{\mathcal{E}_{i+1}} \mid i \in \mathbb{Z}, 0 \leq i < n\}$. Let $\mathcal{K}' = (\mathcal{T}, \mathcal{D}')$ and $\mathcal{E}' =$ `ComputeRanking`$(\mathcal{K}')$. Then $\mathcal{E} \doteq \mathcal{E}'$.*

The proof of this uses the following lemmas:

Lemma 5. $\mathcal{T} \models \bigsqcap \overline{\mathcal{E}_0'} \equiv \bigsqcap \overline{\mathcal{D}'}$ *and* $\mathcal{T} \models \bigsqcap \overline{\mathcal{D}'} \equiv \bigsqcap \overline{\mathcal{E}_0}$.

Lemma 6. *For all* $i \in [1, n]$,

$$\mathcal{T} \models \bigsqcap \overline{\{\neg \bigsqcap \overline{\mathcal{E}_j} \sqsubsetsim \bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i \leq j < n\}} \equiv \bigsqcap \overline{\mathcal{E}_{i+1}}$$

Lemma 7. *For all integers* $i \in [0, n]$,

$$\texttt{Exceptional}(\mathcal{T}, \mathcal{E}_i') = \{\neg \bigsqcap \overline{\mathcal{E}_j} \sqsubsetsim \bigsqcap \overline{\mathcal{E}_{j+1}} \mid j \in \mathbb{Z}, i \leq j < n\}$$

6 Related Work and Conclusions

A recent proposal for dealing with defeasible reasoning in the framework of DLs that is relatively related to the present approach is the one by Bonatti, Sauro and others [3,6]. They use a different kind of semantics, and their decision procedures have good computational results and satisfy different properties w.r.t. the KLM-based approach. Similarly, the proposal by Bozzato and others [7] models defeasible reasoning in DLs using an approach based on the notion of context, hence they use a different semantics and the entailment relation satisfies different properties. Strongly related to the work presented here is instead the work by Giordano and others [27,29,28], that is also built on top of the KLM approach. Pensel and Turhan [37,38] have refined the notion of Rational Closure in order to better exploit the expressivity of some DLs like $\mathcal{EL}_\bot$. The work of Britz and Varzinczak moves instead in the direction of extending the kind of defeasible information that can be expressed in the DL framework, adding defeasible quantifiers and role inclusions [9,13,45].

The results in this paper can be expanded upon in a number of ways. From the point of view of implementation, there is a software tool, *PRO* (*Preferential Reasoner for Ontologies*), that models Algorithm 3 and is available online,[†]. It can be the base for the development of more advanced tools. From the theoretical point of view, the most urgent step would be to add an ABox and enable drawing presumptive conclusions about individuals, as it has already been modelled for the Rational Closure and other forms of closure [18,20,15,27,38]. In Section 5 we have related rankings to rational closure for dense rankings - is there a more natural construction for doing so? Can we do the same for other forms of closure, for example, lexicographic closure? More in general, the results presented here can be further developed and analyzed in the light of ongoing work for propositional logic [17], focused on the behaviour of possible refinements of Rational Closure. Also, the results on statement redundancy can be extended to multiple redundant statements.

Acknowledgments. G. Casini and T. Meyer have received funding from the EU Horizon 2020 programme under the Marie Skłodowska-Curie grant agr. No. 690974 (MIREL). The work of T. Meyer has been supported in part by the National Research Foundation of South Africa (grant No. UID 98019).

[†] https://github.com/MindfulMichaelJames/PRO

References

1. Baader, F., Hollunder, B.: How to prefer more specific defaults in terminological default logic. In: Bajcsy, R. (ed.) Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI). pp. 669–675. Morgan Kaufmann Publishers (1993)
2. Baader, F., Hollunder, B.: Embedding defaults into terminological knowledge representation formalisms. Journal of Automated Reasoning **14**(1), 149–180 (1995)
3. Bonatti, P., Faella, M., Petrova, I., Sauro, L.: A new semantics for overriding in description logics. Artificial Intelligence **222**, 1–48 (2015)
4. Bonatti, P., Faella, M., Sauro, L.: Defeasible inclusions in low-complexity DLs. Journal of Artificial Intelligence Research **42**, 719–764 (2011)
5. Bonatti, P., Lutz, C., Wolter, F.: The complexity of circumscription in description logic. Journal of Artificial Intelligence Research **35**, 717–773 (2009)
6. Bonatti, P., Sauro, L.: On the logical properties of the nonmonotonic description logic DL^N. Artificial Intelligence **248**, 85–111 (2017)
7. Bozzato, L., Eiter, T., Serafini, L.: Enhancing context knowledge repositories with justifiable exceptions. Artificial Intelligence **257**, 72 – 126 (2018)
8. Britz, K., Casini, G., Meyer, T., Moodley, K., Sattler, U., Varzinczak, I.: Theoretical foundations of defeasible description logics. Tech. rep. (2019), `http://ijv.ovh/common/papers/DefeasibleDL.pdf`
9. Britz, K., Casini, G., Meyer, T., Varzinczak, I.: Preferential role restrictions. In: Proceedings of the 26th International Workshop on Description Logics. pp. 93–106 (2013)
10. Britz, K., Heidema, J., Meyer, T.: Semantic preferential subsumption. In: Lang, J., Brewka, G. (eds.) Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR). pp. 476–484. AAAI Press/MIT Press (2008)
11. Britz, K., Heidema, J., Meyer, T.: Modelling object typicality in description logics. In: Nicholson, A., Li, X. (eds.) Proceedings of the 22nd Australasian Joint Conference on Artificial Intelligence. pp. 506–516. No. 5866 in LNAI, Springer (2009)
12. Britz, K., Meyer, T., Varzinczak, I.: Semantic foundation for preferential description logics. In: Wang, D., Reynolds, M. (eds.) Proceedings of the 24th Australasian Joint Conference on Artificial Intelligence. pp. 491–500. No. 7106 in LNAI, Springer (2011)
13. Britz, K., Varzinczak, I.: Introducing role defeasibility in description logics. In: Michael, L., Kakas, A. (eds.) Proceedings of the 15th European Conference on Logics in Artificial Intelligence (JELIA). pp. 174–189. No. 10021 in LNCS, Springer (2016)
14. Casini, G., Meyer, T., Moodley, K., Sattler, U., Varzinczak, I.: Introducing defeasibility into OWL ontologies. In: Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K., Staab, S. (eds.) Proceedings of the 14th International Semantic Web Conference (ISWC). pp. 409–426. No. 9367 in LNCS, Springer (2015)
15. Casini, G., Meyer, T., Moodley, K., Varzinczak, I.: Nonmonotonic reasoning in description logics: Rational closure for the ABox. In: Proceedings of the 26th International Workshop on Description Logics. pp. 600–615 (2013)
16. Casini, G., Meyer, T., Moodley, K., Varzinczak, I.: Towards practical defeasible reasoning for description logics. In: Proceedings of the 26th International Workshop on Description Logics. pp. 587–599 (2013)

17. Casini, G., Meyer, T., Varzinczak, I.: Taking defeasible entailment beyond rational closure. In: Proceedings of the 16th European Conference on Logics in Artificial Intelligence (JELIA). LNCS, Springer (forthcoming)
18. Casini, G., Straccia, U.: Rational closure for defeasible description logics. In: Janhunen, T., Niemelä, I. (eds.) Proceedings of the 12th European Conference on Logics in Artificial Intelligence (JELIA). pp. 77–90. No. 6341 in LNCS, Springer-Verlag (2010)
19. Casini, G., Straccia, U.: Lexicographic closure for defeasible description logics. In: Proceedings of the 8th Australasian Ontology Workshop (AOW). vol. 969, pp. 4–15. CEUR Workshop Proceedings (2012)
20. Casini, G., Straccia, U.: Defeasible inheritance-based description logics. Journal of Artificial Intelligence Research (JAIR) **48**, 415–473 (2013)
21. Casini, G., Meyer, T., Moodley, K., Nortje, R.: Relevant closure: A new form of defeasible reasoning for description logics. In: Fermé, E., Leite, J. (eds.) Proceedings of the 14th European Conference onLogics in Artificial Intelligence (JELIA). pp. 92–106 (2014)
22. Casini, G., Straccia, U.: Defeasible inheritance-based description logics. In: Walsh, T. (ed.) Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI). pp. 813–818 (2011)
23. Casini, G., Straccia, U., Meyer, T.: A polynomial time subsumption algorithm for nominal safe $\mathcal{ELO}_\bot$ under rational closure. Information Sciences (2018). https://doi.org/10.1016/j.ins.2018.09.037
24. Donini, F., Nardi, D., Rosati, R.: Description logics of minimal knowledge and negation as failure. ACM Transactions on Computational Logic **3**(2), 177–225 (2002)
25. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.: Reasoning about typicality in preferential description logics. In: Hölldobler, S., Lutz, C., Wansing, H. (eds.) Proceedings of the 11th European Conference on Logics in Artificial Intelligence (JELIA). pp. 192–205. No. 5293 in LNAI, Springer (2008)
26. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.: A non-monotonic description logic for reasoning about typicality. Artificial Intelligence **195**, 165–202 (2013)
27. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.: Semantic characterization of rational closure: From propositional logic to description logics. Artificial Intelligence **226**, 1–33 (2015)
28. Giordano, L., Dupré, D.T.: Defeasible reasoning in $\mathcal{SROEL}$: from rational entailment to rational closure. Fundam. Inform. **161**(1-2), 135–161 (2018)
29. Giordano, L., Gliozzi, V., Olivetti, N.: Towards a rational closure for expressive description logics: the case of $\mathcal{SHIQ}$. Fundam. Inform. **159**(1-2), 95–122 (2018)
30. Governatori, G.: Defeasible description logics. In: Antoniou, G., Boley, H. (eds.) Rules and Rule Markup Languages for the Semantic Web. pp. 98–112. No. 3323 in LNCS, Springer (2004)
31. Grosof, B., Horrocks, I., Volz, R., Decker, S.: Description logic programs: Combining logic programs with description logic. In: Proceedings of the 12th International Conference on World Wide Web (WWW). pp. 48–57. ACM (2003)
32. Heymans, S., Vermeir, D.: A defeasible ontology language. In: Meersman, R., Tari, Z. (eds.) CoopIS/DOA/ODBASE. pp. 1033–1046. No. 2519 in LNCS, Springer (2002)
33. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence **44**, 167–207 (1990)
34. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artificial Intelligence **55**, 1–60 (1992)

35. Lutz, C.: The complexity of conjunctive query answering in expressive description logics. Automated Reasoning pp. 179–193 (2008)
36. Padgham, L., Zhang, T.: A terminological logic with defaults: A definition and an application. In: Bajcsy, R. (ed.) Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI). pp. 662–668. Morgan Kaufmann Publishers (1993)
37. Pensel, M., Turhan, A.Y.: Including quantification in defeasible reasoning for the description logic $\mathcal{EL}_{\bot}$. In: Balduccini, M., Janhunen, T. (eds.) Proceedings of the 14th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR). pp. 78–84. No. 10377 in LNCS, Springer (2017)
38. Pensel, M., Turhan, A.: Reasoning in the defeasible description logic $\mathcal{EL}_{\bot}$ - computing standard inferences under rational and relevant semantics. Int. J. Approx. Reasoning **103**, 28–70 (2018)
39. Qi, G., Pan, J., Ji, Q.: Extending description logics with uncertainty reasoning in possibilistic logic. In: Mellouli, K. (ed.) Proceedings of the 9th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty. pp. 828–839. No. 4724 in LNAI, Springer (2007)
40. Quantz, J., Royer, V.: A preference semantics for defaults in terminological logics. In: Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR). pp. 294–305 (1992)
41. Quantz, J., Ryan, M.: Preferential default description logics. Tech. rep., TU Berlin (1993), `www.tu-berlin.de/fileadmin/fg53/KIT-Reports/r110.pdf`
42. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements. Artificial intelligence **48**(1), 1–26 (1991)
43. Sengupta, K., Alfa Krisnadhi, A., Hitzler, P.: Local closed world semantics: Grounded circumscription for OWL. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) Proceedings of the 10th International Semantic Web Conference (ISWC). pp. 617–632. No. 7031 in LNCS, Springer (2011)
44. Straccia, U.: Default inheritance reasoning in hybrid KL-ONE-style logics. In: Bajcsy, R. (ed.) Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI). pp. 676–681. Morgan Kaufmann Publishers (1993)
45. Varzinczak, I.: A note on a description logic of concept and role typicality for defeasible reasoning over ontologies. Logica Universalis **12**(3-4), 297–325 (2018)